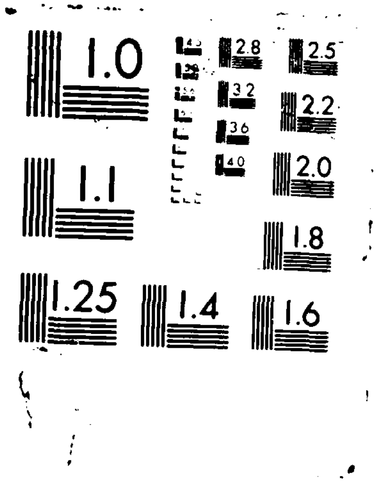
AD-A190 240 FORT LEAVENWORTH IMPROVED KELLNER STATION PACKAGE(U) 1/1
ARMY TRADOC ANALYSIS COMMAND FORT LEAVENWORTH KS
T DANIELS NOV 87 TRAC-F-TM-0687

UNCLASSIFIED F/G 12/5 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

2

Technical Memorandum TRAC-F-TM-Ø687
November 1987

TRADOC Analysis Command-Fort Leavenworth (TRAC-FLVN)
Operations Directorate, Technical Analysis Branch
Fort Leavenworth, Kansas 66Ø27-52ØØ

FORT LEAVENWORTH IMPROVED KELLNER STATION PACKAGE

(FLIK-STN)

by

Tim Daniels

ACN 48722

The views, opinions, and/or findings contained in this report are not to be construed as an official Department of the Army position, policy, or decision unless so designated by authorized documents issued and approved by the Department of the Army.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for Public Release
Distribution is Unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
TRAC-F-TM-0687

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
TRAC-FLVN

6b. OFFICE SYMBOL *(If applicable)*
ATRC-FOC

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Director, TRAC-FLVN
ATRC-FOC
Ft Leavenworth, KS 66027-5200

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*

12. PERSONAL AUTHOR(S)
Mr. Tim Daniels

13a. TYPE OF REPORT
Final Report

13b. TIME COVERED
FROM 6-87 TO 11-87

14. DATE OF REPORT *(Year, Month, Day)*
1987, November

15. PAGE COUNT
23

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED □ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE *(Include Area Code)*

22c. OFFICE SYMBOL

Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

Table of Contents

Abstract

This document is a user's manual for the Fort Leavenworth Improved Kellner Station software package. It explains the basic purpose of that software, explains its relationship to other graphics software, and describes prerequisite activities to using this software. Then, this document instructs the user on installation of this software, and on its operation. This document also explains generally the operation of Ramtek hardware from a programmers' perspective. Finally the document contains a sample installation run and the set of error messages that may occur during installation procedures.

1. Introduction. The subject of this document is the Fort Leavenworth Improved Kellner (FLIK) Station Support Package (FLIK-STN). FLIK itself functions as a graphics package for user applications running with a Ramtek 94xØ controller. The function of FLIK-STN is to maintain and provide to FLIK the data that describes the critical attributes of Ramtek hardware dedicated to specific stations. The purpose of FLIK-STN is to enable FLIK to free user applications from virtually all considerations of peculiarities relevant to specific Ramtek 94xØ controller or stations. The FLIK graphics package will work only after the FLIK-STN package has been correctly installed. Familiarity with Ramtek hardware will greatly facilitate the understanding and use of the material contained in this document.

a. FLIK-STN has two parts: FLIK-STN-INST, installation station software that allows you to describe Ramtek stations, and FLIK-STN-UTIL, a library of subroutines callable by FLIK, to ascertain the previously defined station information.

b. We include FLIK-STN in the distribution tape for FLIK. FLIK distribution includes the directories listed below. Refer to the Appendix A for listings of these directories.

- [FLIK] with command file ØKASG.COM to establish logicals correctly for FLIK and related software

- [FLIK.STATION] contains this FLIK-STN software, both FLIK-STN-INST, for installation and FLIK-STN-UTIL, the library of callable modules

- [FLIK.MARQUIS] containing that portion of the Marquis software which is legal to distribute.

- [FLIK.FLIK_V2] version two of FLIK graphics text, object, and help libraries plus common block INCLUDE files.

- [FLIK.FLIK_V2.MODULES] documentation: two files for each functional category of FLIK: one containing general documentation and the other containing a list of all FLIK modules in that functional area. These categories stem from chapter titles in the Ramtek RM-94ØØ Series Software Training Manual.

- [FLIK.FLIK_V2.PROGRAMMER] documentation: programmer reference manual, printable, organized into the same functional areas.

- [FLIK.STRINGS] for string manipulation and other utilities required with the various FLIK or supporting software.

- [FLIK.OMNI] separate software package for a Ramtek mnemonic graphics package used as a teaching tool, made to duplicate Ramtek's OMNI at their software class.

c. Note that you may use this FLIK-STN package with any in-house developed graphics package that works with Ramtek hardware. FLIK-STN works with VAX (Digital Equipment trademark) host processor, using the VMS (Digital Equipment trademark) operating system, version 4.1 and up, with Ramtek 9400/9460 series graphics hardware, and with the Marquis device driver. Any deviations from the specified operating system, graphics hardware, or hardware driver is at your own risk.

2. POC. The point of contact for any questions on FLIK-STN or FLIK, TRAC OMNI, or any other TRAC-FLVN graphics software mentioned in this document is

Director, TRAC FLVN
ATRC-FOC-T (Tim Daniels)
Fort Leavenworth, KS 66027-5200

or phone AV 552-4392/3193/3617/5258.

Also refer to the following documents:

- Ramtek 9400/9460 System Description Manual, product number 8000102-02A

- Ramtek RM-9460 Graphics Display System Software Reference Manual product number 8000081-02

- Ramtek RM9400 Series Software Training Manual, no product number

- Ramtek VAX/VMS RM-9400/9460 Device Driver Reference Manual, product number 8000085-01A

- Ramtek RM9400 Graphics Display System Hardware Reference Manual, product number 504646-01

3. Overview. Here we present the purpose for FLIK-STN, how it services FLIK, and where it fits in the overall graphics software system.

a. Troubles arise when running on one Ramtek with software that was designed for another Ramtek configuration. This FLIK-STN package solves that hardware compatability problem by maintaining data for each Ramtek station which it passes to FLIK, upgraded to resolve conflicts between the user's expected Ramtek characteristics and those of the Ramtek hardware actually used.

b. From the top down, user applications software runs on the host computer and draws graphics by calling FLIK. When called, FLIK, also on the host, translates those graphics calls to Ramtek instructions which it then sends to the Ramtek hardware. On receipt of those instructions, the firmware resident on the Ramtek hardware parses them and responds with the appropriate graphics actions.

c. The hardware compatability problem solved by FLIK-STN occurs in two ways during translation of graphics calls to Ramtek instructions. First, certain characteristics of that Ramtek hardware are very critical, such as the number of refresh memory planes in a group, digital to analog convertor (DAC) size on the video board, type of keyboard, type of peripheral, the overall signal path through the Ramtek from host to picture and refresh memory resolution. Second, the specification of hardware to use either during certain instructions or until respecified later becomes station-dependent. Each Ramtek chassis can act as several Ramtek stations, assigning to each station certain duplicate hardware or portions of boards, including duplicate memory control processor (MCP), memory group (MG), video board, keyboard, peripheral, and cursor generator. The Ramtek firmware does not define a "station" or its constituent parts, as such, but it does allow FLIK to specify the use of that hardware. Upgrading FLIK's translation logic to use the Ramtek station descriptions supplied to it by FLIK-STN-UTIL virtually solves the hardware compatability problem.

d. Both parts of FLIK-STN, plus the FLIK upgrades, provide a graphics environment wherein the user application functions virtually independent of Ramtek peculiarities or station selected.

4. Preparation. Here we present the general description and order of the tasks required to prepare FLIK and FLIK-STN for use by application programs. We intend that FLIK and FLIK-STN reside on a VAX hard disk, and we allow for the possibility that that disk drive may be accessed through a server by more than one central processing unit (CPU). The distribution media for FLIK and FLIK-STN is magnetic tape. You do most of these preparation steps only once on a given disk. However, you do some of these steps once for each CPU that can access the FLIK and FLIK-STN disk and that has Ramtek controllers attached to it.

a. Marquis device driver. Prior to any work with FLIK or FLIK-STN, the system manager must have already installed the Marquis driver, and the Marquis driver FORTRAN support package, as per instructions in the Marquis documentation. This software is not actually a part of the FLIK distribution.

b. FLIK tape. Prior to any work with FLIK-STN, the system manager must install the distribution of FLIK, using the VAX VMS DCL (Digital Equipment Trademark) BACKUP command shown below to roll FLIK, FLIK-STN, and associated software from the distribution tape onto disk.

```
$ BACKUP /LOG/VER/REW <tape>FLIK.BCK <disk>[FLIK...]
```

c. "Logicals." Prior to any work with FLIK-STN, the system manager must modify [FLIK]ØKASG.COM to provide all necessary logicals for FLIK, FLIK-STN, and the Marquis software, according to the FLIK installation guide found on [FLIK]INSTALL.DOC. These logicals include the following.

LOGICAL NAME	CPU DEPENDENT	PACKAGE REFERENCED
ØKGL	NO	FLIK
ØRGL	NO	MARQUIS FORTRAN PKG
ØRMSTN	NO	FLIK-STN
ØRMSTNDAT	YES	EACH CPU'S STATIONS

d. FLIK-STN installation. Now you work with FLIK-STN-INST software, described later in paragraph five, Installation. This is the installation phase of FLIK-STN, the software described in this document. For each CPU with Ramteks attached, graphics personnel should run ØRMSTN:STATION.EXE, the interactive portion of FLIK-STN to create one station description file, and then rename that file to STATION.DAT in the subdirectory for that CPU under [FLIK.STATION]. Paragraph five, Installation, contains a complete "how-to" for ØRMSTN:STATION.EXE, plus a sample run and a sample "batch" input file.

e. FLIK-STN utilities. After FLIK-STN installation, the FLIK-STN-UTIL software will function correctly. FLIK-STN-UTIL comprises the second portion of FLIK-STN, the software described in this document. FLIK graphics already contains calls to FLIK-STN-UTIL subroutines to read the station descriptor file, to identify the current station, and to read back information about that station from the descriptor file.

5. Installation. Here we present the installation program, ØRMSTN:STATION.EXE, the FLIK-STN-INST portion of FLIK-STN software. The description includes its functions, preparation for use, a "How-To" on its use, a sample run, and a sample "batch" input. The tasks to perform prior to running this software appear in paragraph four, "Preparation." These include rolling the FLIK distribution from tape to disk, installing the Marquis software, and modifying the command file that assigns logicals for use by graphics software. The goal of the installation is to produce one station-descriptor file for each CPU that uses Ramtek hardware. Each run of ØRMSTN:STATION.EXE works with one such file. Eventually each station description file must reside in ØRMSTNDAT, the CPU-dependent directory. In that way, software running on any CPU can access the station file made for that CPU under ØRMSTNDAT:STATION.DAT.

a. Functions. Each run of ØRMSTN:STATION.EXE works with one new or existing station descriptor file, which you identify at the start of the program. If you select an existing file, you leave it intact and create a new version; you may also read all existing station descriptions for possible transfer to the new file version, with or without modification.

(1) As mentioned in paragraph four, Preparation, you need

a subdirectory under [FLIK STATION] for each CPU that can access the FLIK disk and that uses Ramteks. For example, one might use [FLIK.STATION.V78Ø] and [FLIK.STATION.V86ØØ]. Then, you put into each of those subdirectories a file named STATION.DAT. You use one run of ØRMSTN:STATION.EXE to create each STATION.DAT file.

(2) A Ramtek station consists of the hardware necessary for one person to interface with a computer, via graphics. This set of hardware includes a color monitor, a tablet, light pen, mouse, or cursor controller, perhaps a keyboard, and all or part of a Ramtek controller chassis. It is quite common for one such chassis to support more than one Ramtek station, using duplicate boards for the duplicate functions of each station served.

(3) After you select a station descriptor file, ØRMSTN:STATION.EXE has several functions:

- View all station descriptions already on the station file.

- Select a station, new or existing.

- View an existing station's description.

- Describe a new station or redefine an existing one, either from the terminal or with input from a "batch" file. This description is vulnerable: it is only preserved on disk when you save the entire file later.

- Save the station descriptions on a new file, or on a new version of an existing station file.

(4) After a complete installation for one CPU, the contents of that file must describe all stations served by all chassis connected to that CPU. The FLIK graphics package tells FLIK-STN-UTIL, the utility portion of FLIK-STN, to read its station information from ØRMSTNDAT:STATION.DAT. You must make one of these files for each CPU that uses Ramtek hardware. Since the files names are all identical, you distinguish them by the directory that contains them. You must put each of the station files that you create on the proper subdirectory under [FLIK.STATION] set up previously for the CPU whose stations are described on that file. For example, if you have one VAX 86ØØ using Ramtek hardware, during the steps prior to running STATION.EXE, you should have created a subdirectory for that V86ØØ, say [FLIK.STATION.V86ØØ], and altered [FLIK]ØKASG.COM to assign ØRMSTNDAT to that subdirectory for anyone logging on to the V86ØØ. Now, after making the station file for that CPU, you rename it as [FLIK.STATION.V86ØØ]STATION.DAT. Subsequently, when users log on to the V86ØØ and run graphics using FLIK, they will access ØRMSTNDAT:STATION.DAT which translates to [FLIK.STATION.V86ØØ]STATION.DAT, the file you just created for the stations on the V86ØØ.

b. Assistance. Note that the more you know about Ramtek hardware and graphics, the easier it is for you to run STATION.EXE. This program requires careful answers to very detailed questions about the Ramtek hardware, which may seem very discouraging at first. But there is help available. STATION.EXE does contain several features that will enable you to get up and running in short order with a significant degree of success. We urge you to take advantage of these features.

(1) Most of the prompts in this installation program have a Help option that will give you background information about the requested input. Also, STATION.EXE follows many of the prompts with square brackets containing a valid range or set of inputs, and subsequent square brackets containing a default input value. These ranges serve as a guideline for input, and the defaults, entered with a blank and a carriage return, are a convenience. STATION.EXE will use these default values if you wish to create a valid station description. Be careful that when you use these default values the resulting description fits your Ramtek hardware.

(2) In addition, the alternative "batch" input mode of STATION.EXE reads station descriptions from an input file instead of from your terminal. This distribution includes several sample "batch" input files under [FLIK.STATION]*.DES, for Ramtek hardware stations at TRAC-FLVN. You may use those files as a guideline for your own terminal input, or you may create modified versions of them to describe the Ramtek stations at your own site, and then run them through this STATION program as "batch" input.

(3) We recommend several sources of information in preparing to answer the prompts in STATION.EXE. In this manual, read paragraph seven, Hardware, for a functional description of the Ramtek hardware. Read the model and configuration number on the label at the front or rear of the Ramtek chassis, which you can use as a reference during calls to the Ramtek home office for details that they retain concerning your hardware configuration. Refer frequently to the system configuration block diagrams in the Ramtek RM-9400 Series Software Training Manual and in the Ramtek 9460/9465 System Description Manual. Correlate your model number with these diagrams that show the various 94x0 models, including backplane, board options, and interconnection. Consult with your hardware engineer who can examine the hardware description from Ramtek that should have accompanied your equipment. As a last resort, refer to paragraph two, POC.

(4) Beware of any field modifications to hardware that Ramtek may not know about. To verify any unknowns about your hardware, refer to your Ramtek software reference manual, and then write tests in OMNI to send instructions to the Ramtek to verify your information. OMNI is totally separate from FLIK and FLIK-STN., and is a a Ramtek-specific mnemonic graphics language available from TRAC-FLVN. It is included with the FLIK

distribution. and it virtually duplicates the functions of the real OMNI, a set of undistributed software written in "C" at Ramtek that is used as a teaching tool in their RM94XØ software course.

c. Preparation. Before you run STATION.EXE to describe the stations on each chassis on a given CPU, we recommend that you determine the exact Ramtek boards or portions thereof that constitute each station and determine the critical attributes thereof.

(1) Refer to paragraph seven, "Hardware", for a discussion of Ramtek boards required by each station.

(2) Much of the information you need to determine the hardware that comprises a given station resides in the configuration data. Correlate this information with block diagrams for your Ramtek model found in both the Ramtek 946Ø/9465 System Description Manual, and the Ramtek 94ØØ Series Training Manual. That will tell you which MCPs, which MGs, which memory planes in those MGs, which peripherals and keyboards, and which video board services a given station. It will also give you the necessary parameters for those boards, such as DAC size and type of peripheral.

(3) Refer to paragraph 5b, Installation Assistance", for suggestions on information sources.

d. Dialog. This is the dialog of the execution of program STATION.EXE. It describes all phases of the program and, for each phase, provides an exact copy of all prompts, along with your input options and how the computer will respond to them. Paragraph nine, Errors, contains an exact copy of any error messages that might arise. Some phases of STATION are iterative, which is noted therein. Also included in this document in paragraph eight, Sample Installation, is a sample run of STATION, and a sample "batch" input file associated with that run.

(1) You start at the DCL level by entering this DCL command:

```
$ RUN ØRMSTN:STATION
```

(2) WARNING: Remember at the end of the program to SAVE the data you enter, else you will lose it all.

(3) Note that STATION.EXE allows options not shown in the prompt. Virtually all prompts will take a response of "?" to ask for help. Within the major section for station description, all of the prompts display a default value that you can select by entering a blank then a carriage return. To exit any iterative phase that permits an early exit, you enter just a carriage return.

(4) In the following dialog description, the prompts written to your screen by the computer are highlighted in boldface type.

FILE? Enter station-data file-spec, or <CR> to quit

This determines the station descriptor file used during one complete run of the STATION.EXE program. No data is automatically saved.

- If you enter the name of a new file, then STATION.EXE creates the file immediately.

- If you enter an existing file name, then the program does not modify this old file but will eventually create a new version.

- If you enter a carriage return, the program will stop.

MODIFY existing file? [Y/N] [N]

This only occurs if you have identified an existing descriptor file.

- If you enter a "Y", the computer reads the current station descriptions from the old file, and will allow you to optionally change them before writing to a new version of that file.

- If you enter any other response, the computer will return to the FILE? prompt.

VIEW FILE? Enter [Y/N] to view all current stations. [N]

If you enter a "Y", the computer shows on the CRT all data for all stations on all chassis currently described on the specified station file. For any other response, the computer does nothing.

BATCH? batch-input file-name, or <SP> for terminal input

- If you enter a correct batch file name, usually one of the *.DES files under [FLIK.STATION], the computer will read all records on the specified file, treating them as station-description input. Then the program will return to this same prompt. See paragraph 5e, Batch Mode, for more information.

- If you enter <SP> <CR>, the computer will take station descriptions from your terminal. After you finish your station descriptions, the program will return to this same prompt.

- If you enter a question mark, the computer gives you information on this script from the associated help library.

- If you enter <CR>, the computer jumps out of this phase and enters the save phase of the program.

STATION? Enter VMS logical like RMAØ:

This section repeats each time you select another station. You control when the repetition stops.

A Ramtek station is part or all of a specific Ramtek controller chassis, and on a VAX, has a unique logical name. You must reference this station by that correct logical name. Consult the system manager for the correct logicals.

- If you enter the logical for a new station, the computer jumps past the next prompt to the MCP? prompt. If you specify the logical for an an existing station, the computer goes to the redefine-station script.

- If you enter <CR>, the computer returns to the BATCH? prompt.

REDEFINE? Enter [Y/N] to redefine existing station. [N]

If the specified station is already defined, by this or an earlier run of STATION.EXE, then the program will try to safeguard that information. First you see the existing data, then you have the chance to modify it.

- If you enter a "Y", the computer will allow you to redefine the description for this station.

- If you enter any other response, STATION.EXE will skip the station description phase and will return to the STATION? prompt.

MCP? Enter Ø for a model 94ØØ, or quantity of MCP's

This generally defines the type of Ramtek chassis for this station. the older RM94ØØs are always a single MCP chassis, while the newer RM946Øs can accommodate one or more MCPs.

- If you enter a zero, the computer assumes an RM94ØØ which may have multiple MGs but for certain only one MCP.

- If you enter a valid positive integer, the computer takes that as the quantity of MCPs on a newer Ramtek chassis.

- If you enter <SP> <CR>, the computer supplies the default value shown after the prompt.

- If you enter a "?", the computer prints help information to your screen.

- You cannot exit this phase with <CR>.

SLC's? Enter quantity on chassis@@

SLC is a mnemonic for "serial link card". All stations on the same chassis use this same information. If a station is already defined on this chassis, then there will already be current information on the number of SLCs, which will appear in the default field after the prompt. Ramtek installs this SLC(s) on the backplane, recorded in the configuration information.

- If you enter a single valid integer, the computer records that value as the number of SLCs on the chassis housing that station.

- If you enter a "?", you receive help on your terminal screen.

- If you enter a <SP> <CR>, the computer keeps the current number of SLCs shown in the default field of the prompt line.

- You cannot leave this phase with a <CR> response.

VIDEO BOARD? Enter board number

This is the video board that has R/G/B cables connected to the monitor at this station. Most likely, your hardware engineer ran that cabling.

- If you enter a zero or positive integer, within the range limit indicated, the computer saves that valid video board number as the one used for this station.

- If you enter a "?", the computer shows you help on your terminal screen.

- If you enter a <SP> <CR>, the computer takes the default value shown after the prompt as the video board number for this station.

- You cannot leave this phase with a <CR> response.

DAC? Enter size in bits

This value depends on the type of video board and strapping therein -- installed by Ramtek, recorded in the configuration information.

- If you enter a single valid integer, from among the valid inputs indicated, the computer saves this as the DAC size on the video board.

- If you enter a "?", the computer shows you help on your terminal screen.

- If you enter a <SP> <CR> input, the computer takes the default value shown after the prompt as the DAC size for the video board at this station.

- You cannot leave this phase with a <CR> response.

VIDEO LINEs? Enter one active line number

This prompt appears repeatedly until you stop it by entering only a <CR>.

The active lines depend on the type of backplane and the number of SLCs -- installed by Ramtek, and recorded in the configuration information.

- If you enter an integer, the computer saves that number as one of the active video lines connected to this station's video board.

- If you enter a "?", the computer gives you help information.

- If you enter a <SP> <CR>, the computer saves the displayed default number as one of the active video lines connected to this station's video board.

PERIPHERALs? Enter one <type> and <dev#> as (2I1)

You enter one or more peripherals, until you stop this phase by entering a <CR>, or until the program reaches a Ramtek hardware limitation.

These data reflect the type of peripherals, which SLC slots they connect to, and settings of switches on the SLCs -- all done by by Ramtek, and recorded in the configuration information.

- If you enter an integer-pair as (2I1 FORMAT), you define peripheral(s) for the current station.

- If you enter a "?", the computer gives you help on this phase.

- If you enter a <SP> <CR>, you take the displayed default value as the descriptor of the peripheral at this station.

As an example, if a station uses a tablet, device-number zero, you would enter the two digits "2Ø".

KEYBOARDs? Enter one <type> and <dev#>, as (2I1)

You enter one or more data elements, until you stop this phase by entering a <CR>, or until the program reaches a Ramtek hardware limitation.

These data reflect the type of keyboard, and which SLC slots they connect to -- all done by by Ramtek, and recorded in the configuration information.

- If you enter an integer-pair as (2I1 FORMAT), you define keyboards(s) for the current station.

- If you enter a "?", the computer gives you help on this phase.

- If you enter a <SP> <CR>, you take the displayed default value as the descriptor of the keyboards at this station.

As an example, if a station uses an old-style Ramtek keyboard, device-number zero, you would enter the two digits "1Ø".

VLT? Enter source as [M/V/B] <qualifier>, bit

VLT is a mnemonic for "video lookup table". In this phase you must enter responses until the program reaches a Ramtek hardware limitation. There is no early exit from this phase.

o You define how this station builds the majority of the final picture, colorized by VLT random access memory (RAM), in terms of the signal paths through hardware to each VLT address bit which receives either picture data, cursor data, or no data. This is primarily for display of the main picture.

o There is a generic path for picture data and another for cursor data. Ramtek defines for each VLT address bit the type of data driving it and the specifics of that data path, using the backplane type, slot locations of MCP, memory, video boards, and SLCs, and the strapping and DIP switches options on the video boards -- all of which they record in the configuration information.

o If there is picture or cursor data "driving" the current VLT bit, then you enter its source as a "type" having possible "qualifiers." Note that the "qualifiers" depend on the "type." The "type" and the associated "qualifier" field(s) appear in the following table.

FOR	---- ENTER THESE FIELDS----		IN THIS FORMAT
TYPE AND SOURCE	ALPHA SOURCE TYPE	INTEGER SOURCE QUALIFIER(S)	FORTRAN INPUT FORMAT
* 1. PICTURE refresh	M	<MCP#> <MG#> <PLANE#> [Ø-7] [Ø-7] [Ø-15]	(A1,I1,I1,I2)
2. CURSOR video-line	V	<VIDEO-LINE#> [based on # of SLCs]	(A1,I1)
3. blink blink-line	B		(A1)

* = THE USUAL SOURCE FOR VLT bits

- If you enter a "?", the computer shows you help on this command.

- If a bit is not driven, enter <CR>. If you want the displayed default source, enter <SP>.

o As an example, if the current bit gets its signals from refresh memory plane number six in MG zero driven by MCP two, then you enter alpha source type "M", and integer source qualifier 2ØØ6, as one response like this: M2ØØ6.

o As another example, if the current bit gets its signals from video line three, then you enter alpha source type "V", and integer source qualifier 3, as one response like this: V3.

PROM? Enter source as [M/V/B] <qualifier>, bit

PROM is a mnemonic for "programmable readonly memory". In this phase, you must enter responses until the program reaches a Ramtek hardware limitation. There is no early exit from this phase.

o You define how this station builds that small portion of the final picture, colorized by overlay PROM, in terms of the signal paths through hardware to each VLT address bit which receives either picture data, cursor data, or no data. This is primarily for cursor displays.

o There is a generic path for cursor data and another for picture data. Ramtek defines for each PROM address bit the type of data driving it and the specifics of that data path, using the backplane type, slot locations of MCP, memory, video boards, and SLCs, and the strapping options on the video boards -- all of which they record in the configuration information.

o If there is cursor or picture data "driving" the current PROM bit, then you enter its source as a "type" having possible "qualifiers." Note that the "qualifiers" depend on the "type." The "type" and the associated "qualifier" field(s) appear below in the following table.

FOR	---- ENTER THESE FIELDS----		IN THIS FORMAT
TYPE	ALPHA	INTEGER	FORTRAN
AND	SOURCE	SOURCE	INPUT
SOURCE	TYPE	QUALIFIER(S)	FORMAT
* 1. CURSOR video-line	V	<VIDEO-LINE#> [based on # of SLCs]	(A1,I1)
2. PICTURE refresh	M	<MCP#> <MG#> <PLANE#> [Ø-7] [Ø-7] [Ø-15]	(A1,I1,I1,I2)
3. blink blink-line	B		(A1)

* = THE USUAL SOURCE FOR PROM bits

- If you enter a "?", the computer shows you help on this command.

- If a bit is not driven, enter <CR>. If you want the displayed default source, enter <SP>

o As an example, if the current bit gets its signals from refresh memory plane number nine in MG zero driven by MCP two, then you enter alpha source type "M", and integer source qualifier 2ØØ9, as one response like this: M2ØØ9.

o As another example, if the current bit gets its signals from video line zero then you enter alpha source type "V", and integer source qualifier Ø, as one response like this: VØ.

SAVE data onto new version of file? [Y/N] [N]

The computer performs this phase when leaving the STATION.EXE program and if you had selected an existing station file

- If you enter any response except "N", the program will save your changes to existing stations, and any new stations that you also added to the old file. It will save all these data, plus any unchanged old station data, on a new version of the old file.

- If you enter a "N", you lose all your changes and additions to the existing station file, and save nothing.

e. Batch mode. The interactive STATION program will also work in an alternative "batch" mode.

(1) First, you perform the normal startup procedures, to select a station descriptor file to create or modify, and optionally view any preexisting stations. This is the point where you may begin the alternative "batch" input mode. You do

so at the BATCH? prompt by entering a "batch" input file name, instead of <SP> <CR>.

(2) The input from the "batch" file will echo on the terminal screen. On the "batch" input file, as with terminal input, you can use <SP> <CR> for default inputs, and can use <CR> to exit an input phase or to indicate "not driven" for VLT or overlay PROM bits.

(3) WARNING. When running with a "batch" file, containing descriptions of existing stations, the program will not ask you for those stations if you want to change the description. It will just go ahead and do it. This is different from the interactive terminal mode, where there is a safeguard against that happening. If you do inadvertently change an existing station description, you can revert to the old version of the station file, which this program preserves.

(4) To facilitate the use of STATION.EXE, you can prepare "batch" files for stations at your own site. You can do so from scratch, or copy and modify a sample "batch" input file contained in the FLIK distribution under [FLIK.STATION...]*.DES. We suggest a naming convention for these batch files to reflect the site, chassis, or station described.

6. Operation. Here we present how to link and run a user graphics application with FLIK and FLIK-STN.

a. Before you link and run applications programs that draw graphics, someone must have already performed all of the steps described in paragraph four, Preparation, and paragraph five, Installation. We intend that FLIK and FLIK-STN reside on a VAX hard disk, accessed by one or more CPUs. Most of the preparation steps should be preformed only once for each disk containing FLIK and FLIK-STN, while the installation phase should be performed once for each CPU that can access the FLIK disk and that has Ramtek controllers attached.

b. After someone does those initial steps, you need only to do three or four things to make a user application work with FLIK and FLIK-STN: invoke [FLIK]ØKASG.COM, optionally link a new executable, then run the user application, and specify the Ramtek station to work with.

c. To invoke the command file containing logicals for Marquis, FLIK, and FLIK-STN software, enter this DCL command or add it to your login procedure. Watch out for the disk drive name.

```
$ @[FLIK]ØKASG.COM
```

d. To link an executable made from the user applications software, FLIK, FLIK-STN, and the Marquis support software package, do a link command similar to the following one.

```
$ @<disk>[FLIK]ØKASG.COM
$ LINK -
<application.obj>,
ØKGL:FLIK.OPT
```

e. Run an executable made with the above DCL commands in the usual way, as follows.

```
$ RUN <application.exe>
```

f. To specify the Ramtek station to work with, while the program runs, its graphics initialization phase uses FLIK to ask you for, and to receive, the logical name for the Ramtek station to use. such as RMA1:. With a correct logical name, the initialization succedds and your application program can then use FLIK to draw on the Ramtek graphics including lines, points, text, polygons, circles, arcs, and to read from tablet, set colors, etc.

g. We discuss further the FLIK initialization to emphasize the necessity of the upcoming list of seemingly minor tasks to perform before running a user program. In detail, the FLIK initialization module is KRMINIT. The primary features of KRMINIT are as follows. KRMINIT asks you for, and waits until it receives, a VAX VMS logical name for a Ramtek station. KRMINIT opens a VAX channel to that station. KRMINIT then uses FLIK-STN-UTIL modules to open and read all contents of the station file that describes all Ramtek stations on the CPU you are presently on. This file has a hardwired name of ØRMSTNDAT:STATION.DAT, including the logical ØRMSTNDAT:. This is a CPU-dependent logical. Next, KRMINIT uses FLIK-STN to access important information about that station, read from the ØRMSTNDAT:STATION.DAT file. This station's description then becomes available through other FLIK-STN-UTIL modules to FLIK, when needed in translating user's graphics requirements to those that will work on the current station.

h. In summary, prior to linking a graphics executable, or running user graphics programs on a selected Ramtek station, these things must have been done right. FLIK and FLIK-STN software must be on disk. The command file [FLIK]ØKASG.COM must contain the correct logicals for access to Marquis, FLIK, and FLIK-STN. You must invoke it during each login. The FLIK-STN installation must have already been completed once for each CPU that can access the FLIK disk and that has Ramtek controllers attached. It is also essential that the Ramtek logical station names match, as found in three widely separated places. These three places are: the logical device name in the DCL $CONNECT command, included in the Marquis portion of the system startup file; the Ramtek station logical name, included in the input to the FLIK-STN installation program ØRMSTN:STATION.EXE; the logical name input to FLIK's KRMINIT module, during initialization of a user application.

7. Hardware. A graphics station is a suite of hardware that serves as a visual computer interface with one individual. A graphics station provides one individual picture on a screen, and one set of devices through which the person can respond to the system. The heart of a Ramtek station is the graphics controller chassis which connects to a larger computer such as the VAX. One Ramtek controller chassis may comprise more than one graphics station.

a. System Overview. The Ramtek 94x0 graphics system receives commands from the host and draws pictures on one or more monitors. It also passes information received from tablets, joysticks, trackballs, keyboards, etc, back to the host computer. To perform these tasks requires a firmware and a hardware system of several major functional elements. Each one of these functional elements generally resides on a distinct type of circuit board, such as display controller, MCP, MG, video, SLC, etc. Each Ramtek controller chassis has a backplane with interconnected slots for these various boards. One Ramtek controller can be made to serve multiple stations primarily by duplicating hardware boards of various types on the backplane. These duplicate boards perform the same functional elements in different stations. For such a configuration, the host must keep a record of which boards belong to which station. That is the primary purpose of the FLIK-STN described in this document. Below, we explain the function and relation of the major boards.

(1) Working backward through the Ramtek controller, the pictures displayed by a monitor come from an R/G/B video signal produced in the Ramtek by a video board that repeatedly reads that picture from a refresh memory board(s). That memory is arranged in a two-dimensional Cartesian coordinate system. There is a memory location for each screen pixel. The overall memory dimensions equal the screen width and height, in pixels. This picture data in refresh memory usually comes from an MCP board, responding to graphics drawing commands. This MCP honors graphics commands sent by the single display control processor board. The display control processor receives commands sent by the host. The format of these commands is documented in the Ramtek RM9460 Graphic Display System Software Reference Manual. It is those standards to which FLIK must conform to make the Ramtek function correctly.

(2) Working forward through the system, the Ramtek commands from the host enter the display controller that translates them to MCP commands. Those MCP commands enter the MCP that draws pictures in refresh memory. The refresh memory picture data enters the video board to be colorized and converted to R/G/B video signals, which go outside the Ramtek controller to the monitor, where you see the final picture displayed.

(3) There are various relationships between the quantity of elements that interconnect in this system. Working backward, each monitor has R/G/B input from one video board. Each video board reads from one or more MGs. Each MG receives its picture from only one MCP. When a video board reads from several MGs those MGs may each receive their picture data from different MCPs. All MCP'S get their commands from the single display controller. Working forward again, one display controller drives all MCPs on the chassis. One MCP can drive more than one MG on a RM9400 controller.

(4) For a single station, most of the necessary hardware functional elements contribute to graphics displays, and include the display controller, at least one MCP, at least one MG per MCP, and only one video board whose R/G/B output connects to the monitor at that station. For the user's response capability, through peripherals and keyboards, the Ramtek controller usually has at least one SLC to which you connect these devices. Devices for more than one station can connect to the same SLC. If the controller chassis has no SLC installed, then the display controller will handle only one peripheral and one keyboard, total, for all stations on the chassis.

b. System parameters. Critical to the proper operation of FLIK are some of the attributes and parameters of, and the backplane interconnections among, these major function hardware boards. For example, the video board has a VLT that determines the color and data hierarchy of the pictures in the refresh memory. These are high level graphics functions that depend on a subsystem on the Ramtek. The characteristics of the VLT and how it is addrussed by refresh memory become quite important.

c. Hardware board types. Each of the paragraphs below discusses hardware board functions and their parameters that you must set in the station description file. You do so during the install phase, while running STATION.EXE.

(1) SLCs communicate with peripherals, and generate cursor picture signals on backplane video lines. Your controller can work without them, but may have one or two. If your chassis has no SLCs, then the display controller replaces its functions, on a limited scale. SLC performs these main functions.

- Input of up to four cursor control or tablet or keyboard devices.

- Processing for four cursor generators,

- Output of four video lines of cursor information.

For cursor data, there are various downloadable software controlled relationships among the input, process, and output functional elements on an SLC. Each cursor controller or tablet drives one or more selected cursor generators. Then each cursor generator connects to only one selected video line.

(2) Video board number. Each video board has a number, from zero to seven. There is always a unique video board for each station. Depending on the backplane, a Ramtek chassis can usually hold more than one video board. A video board converts data from refresh memory, video lines, and blink options, into an analog picture signal that will display on a monitor screen. In doing so, it colorizes or shades that data. The video board data flow for input/process/output is this, generally:

- Input from refresh memory bit-planes, video lines, and blink circuits.

- Process of combining these data, coloring them, and converting to analog video picture.

- Output of a single red/green/blue analog video picture.

3. DAC size. The video board's color R/G/B output uses three DACs, one each for red, green, and blue. The video board's gray-scale output uses one DAC. The RAM VLT provides input to these DACs, and requires a sufficient length for each logical VLT word, based on the DAC size on the board.

- The DAC size is either four or eight bits per color. Four-bit DACs require thirteen-bit VLT words: one blink bit, four red bits, four green bits, and four blue bits. Eight-bit DACs require twenty-five bit VLT words: one blink bit, eight red bits, eight green bits, and eight blue bits.

- The host must set these VLT colors, and thus needs to know the DAC sizes to properly configure the VLT. The older Ramtek systems provide no way to read back this information from the Ramtek firmware, but on the MC68ØØØ based controllers, you may do so with the READ CONFIGURATION instruction.

(4) Video lines. Video lines are circuits on the backplane that carry cursor display signals from some source to a video board that will in turn display cursors on a monitor screen. These video lines receive cursor information either from SLCs, if installed, or from the display controller.

- A backplane usually has eight video lines. Each SLC has four video line outputs to the backplane. In the absence of SLCs, the display controller has only one video line output to the backplane.

- An "active" video line is one that the backplane actually connects to cursor display outputs from installed hardware, either an SLC or the display controller. Therefore, on a specific controller chassis, the backplane type and the SLCs installed determine which video lines are active.

- A video line is useful to a station only when it is "active" and when the backplane connects it to the video board installed for that station. Each video board can have up to four video line inputs from the backplane.

- This chart tells you which lines are "active," based on the number of SLCs on the chassis.

QTY OF	RANGE OF ACTIVE	
SLCs	VIDEO LINE NUMBERS	
INSTALLED	FROM	TO
0	0	0
1	0	3
2	0	7

(5) Peripherals. Peripherals are separate hardware devices at a graphics station that enable a user to enter nonkeyboard responses. These peripheral devices connect to sockets on the Ramtek chassis, either on the SLC(s) if installed, or on the display controller.

- There are various types of peripherals, shown in the following chart relating FLIK-STN's integer equivalent to the known peripheral device types.

TYPE	PERIPHERAL
NUMBER	USED
1	CURSOR CONTROLLER
2	TABLET

- The "device-number" value range depends on the number of SLCs on the chassis, as shown in the following chart. The specific device number depends on which socket on which board you connect the cursor controller or tablet to.

QUANTITY OF	CURSOR OR TABLET		
SLC'S	DEVICE NUMBER RANGE		
INSTALLED	FROM	TO	
0	0	0	
1	0	3	
2	0	3	SLC 1
	4	7	SLC 2

- The first SLC has sockets for devices zero through three. If you have a second SLC, it will have sockets for device numbers four through seven. If you have no SLCs, then you can only use one socket, for device zero, on the display controller.

(6) Keyboards. Keyboards are separate equipment at a graphics station that enable a user to enter keyboard responses. These keyboards connect to sockets on the Ramtek chassis, either on the SLC(s) if installed, or on the display controller. There are various types of keyboards shown in the following chart that relates FLIK-STN's integer equivalents to the known keyboard types.

TYPE NUMBER	KEYBOARD USED
1	OLD RAMTEK KB
2	NEW RAMTEK KB

- The "keyboard-number" value range depends on the number of SLCs on the chassis, as shown in the following chart. The specific keyboard number depends on which socket on which board you connect the keyboard to.

QUANTITY OF SLCs INSTALLED	KEYBOARD DEVICE NUMBER RANGE FROM	TO	
Ø	Ø	Ø	
1	Ø	3	
2	Ø	3	SLC 1
	4	7	SLC 2

- The first SLC has sockets for keyboards zero through three. If you have a second SLC, it will have sockets for keyboards numbers from four through seven. If you have no SLCs, then you can only use one socket, for keyboard zero, on the display controller.

(7) VLT inputs. The video board builds the displayed picture, using the VLT to colorize the main part of its contents. For each pixel, this process involves using the pixel's value not as a color but as an address into the VLT. The color stored at that VLT address becomes the color of that pixel. It is the building of the VLT addressing word that is of concern here. You consider any signal path leading to bit(s) in the VLT address word.

- The number of bits in a VLT address varies, according to the number of logical words in the VLT. This depends on the type of video board where the VLT resides, and also on any possible hardware modifications thereto.

- There are several possible signal paths that may lead into these bits. These paths remain fixed during picture display, and usually are hardware dependent, although on later video boards they can be altered somewhat from the host. The possible signal source for VLT address bits are as follows:

o From an MCP writing to an MG containing a certain memory plane, pixel data goes to this station's video board, then through the patches on the board to the VLT address bit.

o From a video line, driven by a certain cursor generator driven by a certain peripheral, cursor information goes to this station's video board, then through the patches on the board to the VLT address bit. This holds true only for the newer video board "V6B," unless there are hardware modifications to some other video board.

o From the blink circuit through the patches on the video board, blink strobe information goes to the VLT address bit. This also holds true only for the newer video board "V6B," unless there are hardware modifications to some other video board.

- To determine the signal path to each bit, you must consider the type of video board with its patching and nDIP switch settings (and any modifications thereto) the type of backplane (and any modifications thereto), the SLCs installed, and the MCPs and memory planes installed.

- Note that a given bit may or may not be "driven", i.e, on a signal path. If not, this ıs usually due to a VLT depth that is less than the maximum.

(8) Overlay PROM inputs. The video board builds the displayed picture, using the overlay PROM logic to colorize a small portion of its contents. For each pixel position, this process involves using the pixel's value plus cursor information not as a color but as an address into the PROM. The color stored at that PROM address becomes the color of that pixel. On most video boards, this PROM color overwrites the VLT color at that pixel. It is the building of the PROM addressing word that is of concern here. You consider any signal path leaaing to bit(s) in the PROM address word.

- There are several possible signal paths that may lead into these bits. These paths remain fixed during picture display, and usually are hardware dependent, although on later video boards they can be altered somewhat from the host. The possible signal sources for PROM address bits are as follows:

o From a video line (driven by a certain cursor generator driven by a certain peripheral), cursor information goes to this station's video board video line inputs, then through the patches on the board, to the overlay PROM address bit. This is the standard configuration for video line cursor information routing on a station's display.

o From an MCP to an MG containing a certain memory plane, pixel data goes to this station's video board input line, then through the patches on the board, to the PROM address bit. This only applies when a refresh memory plane is hardwired for use as an overlay display.

o From the blink circuit through the patches on the video board, blink strobe information goes to the overlay PROM address bit. This holds true only for the newer video board "V6B," unless there are hardware modifications to some other video board.

- To determine the signal path to each bit, you must consider the type of video board (and any modifications thereto), the type of backplane (and any modifications thereto), the SLCs installed, and the MCPs and memory planes installed.

- Overlay PROM addresses are made usually of eight bits: bits zero through three for overlay planes, and bits four through seven for video-line cursor information. However, this depends on the type of video board used, and on any possible hardware modifications thereto.

- Note that a given bit may or may not be on a signal path. If not, this is usually due to the absence of overlay refresh memory, or the backplane video-line connections to the video board.

8. Sample installations. The following is a sample run of program STATION.EXE, including an actual script, and a copy of the "batch" input file used during that run. The two stations described here having logical names RMAØ: and RMA1: reside on a single Ramtek 946Ø controller chassis. Station RMAØ: uses video board zero, and station RMA1: uses video board one. Both are V7A boards and have four-bit DACs. The first two video lines on the backplane feed both video boards. Each station has one tablet, no keyboards, and uses eight planes of refresh memory. On each station, those eight planes drive the lower eight bits of the V7A video board's VLT.

a. Script. This is the script of the exact interaction between user and program. The user inputs appear after the prompt, enclosed in double quotes (""). If you recreate this script from the terminal, DO NOT enter the double quotes.

```
ENTER "?" for help library access"<CR>"
INSTALL DIALOG begins here
FILE?  Enter station-data file-spec, or <CR> to quit-->"STATION.DAT"
BATCH?  batch-input file-name, or <SP> for terminal input-->"RMA.DES"
STATION?  Enter VMS logical like RMAØ:-->"RMAØ: "
MCP?  Enter Ø for a model 94ØØ, or quantity of MCP's [ 1- 3] [ 1] -->"1"
SLC's?  Enter quantity on chassis [ Ø- 2] [ 1] -->"1"
VIDEO BOARD?  Enter board number [ Ø- 7] [ Ø] -->"Ø"
DAC?  Enter size in bits [ 4, 8] [ 4] -->"4"
VIDEO LINEs?  Enter one active line number [ Ø- 7] [ Ø] -->"Ø"
VIDEO LINEs?  Enter one active line number [ Ø- 7] [ 1] -->"1"
VIDEO LINEs?  Enter one active line number [ Ø- 7] [ 2] -->"<CR>"
```

```
PERIPHERALs?  Enter one <type> and <dev > as (2I1) [2Ø] -->"2Ø"
PERIPHERALs?  Enter one <type> and <dev > as (2I1) [21] -->"<CR>"
KEYBOARADs?  Enter one <type> and <dev >, as (2I1) [1Ø] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit  Ø [MØØØØ] -->"MØØØØ"
VLT?  Enter source as [M/V/B] <qualifier>, bit  1 [MØØØ1] -->"MØØØ1"
VLT?  Enter source as [M/V/B] <qualifier>, bit  2 [MØØØ2] -->"MØØØ2"
VLT?  Enter source as [M/V/B] <qualifier>, bit  3 [MØØØ3] -->"MØØØ3"
VLT?  Enter source as [M/V/B] <qualifier>, bit  4 [MØØØ4] -->"MØØØ4"
VLT?  Enter source as [M/V/B] <qualifier>, bit  5 [MØØØ5] -->"MØØØ5"
VLT?  Enter source as [M/V/B] <qualifier>, bit  6 [MØØØ6] -->"MØØØ6"
VLT?  Enter source as [M/V/B] <qualifier>, bit  7 [MØØØ7] -->"MØØØ7"
VLT?  Enter source as [M/V/B] <qualifier>, bit  8 [     ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit  9 [     ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit 1Ø [     ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit 11 [     ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit 12 [     ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit 13 [     ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit 14 [     ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit 15 [     ] -->"<CR>"
PROM?  Enter source as [M/V/B] <qualifier>, bit  Ø [     ] -->"<CR>"
PROM?  Enter source as [M/V/B] <qualifier>, bit  1 [     ] -->"<CR>"
PROM?  Enter source as [M/V/B] <qualifier>, bit  2 [VØ   ] -->"VØ"
PROM?  Enter source as [M/V/B] <qualifier>, bit  3 [V1   ] -->"V1"
PROM?  Enter source as [M/V/B] <qualifier>, bit  4 [V2   ] -->"<CR>"
PROM?  Enter source as [M/V/B] <qualifier>, bit  5 [V3   ] -->"<CR>"
 chassis            Ø station          Ø video-board          Ø
 bits-per-gun           4
video-lines used
          Ø
          1
DEV, TYPE
   Ø TAB
KB, TYPE
source        <type>     <qualifier>          FORMAT
-------       ----  ------------------    ----------
* refresh-memory  M <MCP#>,<MG#>,<plane#> (A1,I1,I1,I2)
  video-line      V <videl-line#>         (A1,I1)
  blink-circuit   B                       (A1)

  * = usual source to drive VLT bit

If a bit is not driven, enter <CR>.


  Ø MCP                Ø   Ø   Ø
  1 MCP                Ø   Ø   1
  2 MCP                Ø   Ø   2
  3 MCP                Ø   Ø   3
  4 MCP                Ø   Ø   4
  5 MCP                Ø   Ø   5
  6 MCP                Ø   Ø   6
  7 MCP                Ø   Ø   7
source      <type>     <qualifier>         FORMAT
-------      ----  ------------------    ----------
```

```
 refresh-memory  M <MCP#>,<MG#>,<plane#> (A1,I1,I1,I2)
*video-line      V <videl-line#>         (A1,I1)
 blink-circuit   B                       (A1)

  * = usual source to drive overlay-ROM  bit

If a bit is not driven, enter <CR>.


  2 VL                  Ø   Ø   Ø
  3 VL                  1   Ø   Ø
STATION?  Enter VMS logical like RMAØ:-->"RMA1: "
MCP?  Enter Ø for a model 94ØØ, or quantity of MCP's [ 1- 3] [ 1] -->"1"
SLC's?  Enter quantity on chassis [ Ø- 2] [ 1] -->"1"
VIDEO BOARD?  Enter board number [ Ø- 7] [ 1] -->"1"
DAC?  Enter size in bits [ 4, 8] [ 4] -->"4"
VIDEO LINEs?  Enter one active line number [ Ø- 7] [ Ø] -->"Ø"
VIDEO LINEs?  Enter one active line number [ Ø- 7] [ 1] -->"1"
VIDEO LINEs?  Enter one active line number [ Ø- 7] [ 2] -->"<CR>"
PERIPHERALs?  Enter one <type> and <dev > as (2I1) [2Ø] -->"21"
PERIPHERALs?  Enter one <type> and <dev > as (2I1) [21] -->"<CR>"
KEYBOARADs?  Enter one <type> and <dev >, as (2I1) [1Ø] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit  Ø [M1ØØØ] -->"M1ØØØ"
VLT?  Enter source as [M/V/B] <qualifier>, bit  1 [M1ØØ1] -->"M1ØØ1"
VLT?  Enter source as [M/V/B] <qualifier>, bit  2 [M1ØØ2] -->"M1ØØ2"
VLT?  Enter source as [M/V/B] <qualifier>, bit  3 [M1ØØ3] -->"M1ØØ3"
VLT?  Enter source as [M/V/B] <qualifier>, bit  4 [M1ØØ4] -->"M1ØØ4"
VLT?  Enter source as [M/V/B] <qualifier>, bit  5 [M1ØØ5] -->"M1ØØ5"
VLT?  Enter source as [M/V/B] <qualifier>, bit  6 [M1ØØ6] -->"M1ØØ6"
VLT?  Enter source as [M/V/B] <qualifier>, bit  7 [M1ØØ7] -->"M1ØØ7"
VLT?  Enter source as [M/V/B] <qualifier>, bit  8 [      ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit  9 [      ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit 1Ø [      ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit 11 [      ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit 12 [      ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit 13 [      ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit 14 [      ] -->"<CR>"
VLT?  Enter source as [M/V/B] <qualifier>, bit 15 [      ] -->"<CR>"
PROM?  Enter source as [M/V/B] <qualifier>, bit  Ø [      ] -->"<CR>"
PROM?  Enter source as [M/V/B] <qualifier>, bit  1 [      ] -->"<CR>"
PROM?  Enter source as [M/V/B] <qualifier>, bit  2 [VØ    ] -->"VØ"
PROM?  Enter source as [M/V/B] <qualifier>, bit  3 [V1    ] -->"V1"
PROM?  Enter source as [M/V/B] <qualifier>, bit  4 [V2    ] -->"<CR>"
PROM?  Enter source as [M/V/B] <qualifier>, bit  5 [V3    ] -->"<CR>"
 chassis            Ø station          1 video-board          1
 bits-per-gun           4
video-lines used
         Ø
         1
DEV, TYPE
  1 TAB
KB, TYPE
source       <type>    <qualifier>       FORMAT
-------      ---- ------------------    ----------
```

```
* refresh-memory  M <MCP#>,<MG#>,<plane#> (A1,I1,I1,I2)
  video-line      V <videl-line#>         (A1,I1)
  blink-circuit   B                       (A1)

  * = usual source to drive VLT bit

If a bit is not driven, enter <CR>.


  Ø MCP                1   Ø   Ø
  1 MCP                1   Ø   1
  2 MCP                1   Ø   2
  3 MCP                1   Ø   3
  4 MCP                1   Ø   4
  5 MCP                1   Ø   5
  6 MCP                1   Ø   6
  7 MCP                1   Ø   7
source      <type>     <qualifier>        FORMAT
-------      ----  ------------------    ----------
 refresh-memory  M <MCP#>,<MG#>,<plane#> (A1,I1,I1,I2)
*video-line      V <videl-line#>         (A1,I1)
 blink-circuit   B                       (A1)

  * = usual source to drive overlay-ROM  bit

If a bit is not driven, enter <CR>.


  2 VL                 Ø   Ø   Ø
  3 VL                 1   Ø   Ø
STATION?  Enter VMS logical like RMAØ:-->
BATCH?  batch-input file-name, or <SP> for terminal input-->
FORTRAN STOP
```

b. Batch file. This is the "batch" input file used in executing STATION.EXE, as shown above. The file shown here is included in the distribution as [FLIK.STATION]RMAØ.DES. You could have entered the station descriptions at the terminal, also. In that case, you would enter the same lines as you see here, but without the inline comments.

Some of the lines on this sample file are empty: they do not contain any space characters. These lines appear in the example file below as <CR>. There are no requests for "default" values in this example. Had there been, those lines would have appeared here as <SP> <CR>.

```
RMAØ:
1           ! QTY OF MCP'S
1           ! QTY OF SLC'S
Ø           ! VIDEO BOARD #
```

```
4           ! DAC SIZE
Ø           ! VIDEO LINE #
1           ! VIDEO LINE #
<CR>
2Ø          ! PERIPHERAL: TABLET, DEV Ø
<CR>
<CR>
MØØØØ       ! VLT BIT Ø SOURCE
MØØØ1       ! VLT BIT 1 SOURCE
MØØØ2       ! VLT BIT 2 SOURCE
MØØØ3       ! VLT BIT 3 SOURCE
MØØØ4       ! VLT BIT 4 SOURCE
MØØØ5       ! VLT BIT 5 SOURCE
MØØØ6       ! VLT BIT 6 SOURCE
MØØØ7       ! VLT BIT 7 SOURCE
<CR>
<CR>
<CR>
<CR>
<CR>
<CR>
<CR>
<CR>
<CR>
<CR>
VØ          ! PROM BIT 2 SOURCE
V1          ! PROM BIT 3 SOURCE
<CR>
<CR>
RMA1:
1           ! QTY OF MCP'S
1           ! QTY OF SLC'S
1           ! VIDEO BOARD #
4           ! DAC SIZE
Ø           ! VIDEO LINE #
1           ! VIDEO LINE #
<CR>
21          ! PERIPHERAL: TABLET, DEV Ø
<CR>
<CR>
M1ØØØ       ! VLT BIT Ø SOURCE
M1ØØ1       ! VLT BIT 1 SOURCE
M1ØØ2       ! VLT BIT 2 SOURCE
M1ØØ3       ! VLT BIT 3 SOURCE
M1ØØ4       ! VLT BIT 4 SOURCE
M1ØØ5       ! VLT BIT 5 SOURCE
M1ØØ6       ! VLT BIT 6 SOURCE
M1ØØ7       ! VLT BIT 7 SOURCE
<CR>
<CR>
<CR>
<CR>
<CR>
<CR>
```

```
<CR>
<CR>
<CR>
<CR>
VØ          ! PROM BIT 2 SOURCE
V1          ! PROM BIT 3 SOURCE
<CR>
<CR>
```

9. Errors. This is the set of error messages written by software in FLIK-STN, during various phases of the installation software, STATION.EXE.

a. Batch. The "batch" file does not exist. Try again.

b. Station. Invalid VAX VMS logical name. Try again.

c. MCP. Out of range. Try again.

d. SLCs. Quantity of SLC's on a chassis must be either zero, one, or two.

e. Board number. Stay within specified limits on the video-board number.

f. DAC size. Invalid DAC size. Use a valid integer input.

g. Video lines. Invalid active video line. Stay within valid range shown. This is a Ramtek hardware limit.

h. Peripherals. Invalid peripheral TYPE or DEVICE NO. Keep these values within the limitations shown in the chart. Remember the number of SLC's on this chassis.

i. Keyboards. Invalid keyboard TYPE or DEVICE NO. Keep these values within the limits shown in the chart. Remember the number of SLC's on this chassis.

j. Source. Invalid alpha type mnemonic for signal source. Use one of the letters in the table of valid inputs.

k. MCP-MG. Invalid integer for MCP, MG, or PLANE NO. Refer to the ranges below each field in the input chart.

l. VL. Invalid number for VIDEO LINE. This valud must correlate with the numbher of SLC's on the chassis.

DISTRIBUTION LIST

No. Copies

Defense Technical Information Center 2
ATTN: DTIC-TCA
Cameron Station
Alexandria, VA 22314

US Army Library
Army Study Documentation and Information 1
Retrieval System (ASDIRS)
ANRAL-RS
ATTN: ASDIRS
Room 1A518, The Pentagon
Washington, D.C. 20310

US Army Combined Arms Research Library (CARL) 1
ATTN: ATZL-SWS-L
Fort Leavenworth, KS 66027-5000

Director, TRAC-FLVN 1
ATTN: ATRC-FOA (Technical Info Center)
Fort Leavenworth, KS 66027-5200

Director, TRAC-WSMR 1
ATTN: ATRC-WSL (Technical Library)
White Sands Missile Range, NM 88002-5502

Director, TRAC-FLVN 3
ATTN: ATRC-FOC
Fort Leavenworth, KS 66027-5200

END
DATE
FILMED
4-88
DTIC